Designing Privacy-aware Internet of Things Applications

Author Manuscript accepted for publication in Information Sciences (Sep 2019)

Designing Privacy-aware Internet of Things Applications

Charith Perera[a,*], Mahmoud Barhamgi[b], Arosha K. Bandara[c], Muhammad Ajmal[d], Blaine Price[c], Bashar Nuseibeh[c]

[a] *Cardiff University, United Kingdom*
[b] *Claude Bernard Lyon 1 University, France*
[c] *Open University, United Kingdom*
[d] *University of Derby, United Kingdom*

Abstract

Internet of Things (IoT) applications typically collect and analyse personal data that can be used to derive sensitive information about individuals. However, thus far, privacy concerns have not been explicitly considered in software engineering processes when designing IoT applications. The advent of behaviour driven security mechanisms, failing to address privacy concerns in the design of IoT applications can have security implications. In this paper, we explore how a Privacy-by-Design (PbD) framework, formulated as a set of guidelines, can help software engineers integrate data privacy considerations into the design of IoT applications. We studied the utility of this PbD framework by studying how software engineers use it to design IoT applications. We also explore the challenges in using the set of guidelines to influence the IoT applications design process. In addition to highlighting the benefits of having a PbD framework to make privacy features explicit during the design of IoT applications, our studies also surfaced a number of challenges associated with the approach. A key finding of our research is that the PbD framework significantly increases both novice and expert software engineers' ability to design privacy into IoT applications.

Keywords: Internet of Things, Software Engineering, Privacy by Design

1. Introduction

The Internet of Things (IoT) [28] is a interconnected collection of physical objects or *'things'* that have computing, sensing and actuation capabilities, together with the ability to communicate with each other and other systems to collect and exchange data. The design and development process for IoT applications is more complicated than that for desktop, mobile, or web applications

*Corresponding author
Email address: charith.perera@ieee.org (Charith Perera)

for a number of reasons. First, IoT applications require both software and hardware (e.g., sensors and actuators) to work together across many different types of nodes (e.g., micro- controllers, system-on-chips, mobile phones, miniaturized single-board computers, cloud platforms) with different capabilities under different conditions [26]. Secondly, IoT applications development requires different types of software engineers to work together (e.g., embedded, mobile, web, desktop). The complexity of different software engineering specialists collaborating to combine different types of hardware and software is compounded by the lack of integrated development stacks that support the engineering of end-to-end IoT applications.

Typically, IoT applications collect and analyse personal data that can be used to derive sensitive information about individuals. While the misuse of this information can have negative consequences for the individuals concerned, it can also lead to security problems, particularly with advent of new behaviour driven security mechanisms. For example, implicit authentication techniques [33] will grant access to systems based on individual behaviour data collected by IoT systems. This intertwining of security and privacy issues, means that privacy needs to be considered as a key requirement for IoT applications. However, thus far, privacy concerns have not been explicitly considered (despite isolated solutions [42, 41]) in software engineering processes when designing and developing IoT applications. This is in part due to a lack of Privacy-by-Design (PbD) methods for the IoT. Further, the engineering complexities explained above have forced software engineers to put most of their efforts towards addressing other challenges such as interoperability and modifiability, resulting in privacy concerns being largely overlooked. Additionally, a lack of knowledge about the tangible and intangible benefits of privacy practices have contributed to privacy challenges being overlooked [35].

We propose to address this issue by providing systematic guidance to help software engineers develop privacy-aware IoT applications. We build on earlier work [27] which derived a set of privacy guidelines by examining Hoepman's [13] eight design strategies and used them to ***assess*** the privacy capabilities of IoT applications and platforms. This paper integrates these guidelines into a PbD framework that includes a method for applying the guidelines during the IoT application ***design*** process. We go on to evaluate how this PbD framework can help software engineers design a number of example IoT applications.

1.1. Contributions

The primary contributions and the scope of this paper are summarised below:

- We evaluate how a proposed set of privacy guidelines can be used to effectively improve IoT application designs. In support of this, we integrate the guidelines with a method for applying them to propose a PbD framework for IoT applications.

- Our method is uniquely designed to address the challenges associated with IoT systems, such as their heterogeneity and distributed nature. This is a

significant difference from existing PbD frameworks, which focus on more general, high-level principles and design strategies (e.g., [13], [35]).

- We gain insights into how our framework could help software engineers improve their design of privacy aware IoT applications by identifying and applying privacy protecting features into their designs.
- We also explore strengths and weaknesses of our approach as well as challenges in manual application design processes in general. We provide insights on how to address these weaknesses.

It is important to note that we do not claim our PbD framework is better than any previous work, nor do we claim that applying set of privacy guidelines will eliminate all privacy risks. To the best of our knowledge, this is one of the first PbD frameworks that explicitly targets IoT application design challenges. Our aim is to maximise software engineers' ability to be aware of and reduce privacy risks during the design phase. We further elaborate the aims of our PbD framework in Section 4.

1.2. Target Audience

We developed our PbD framework as a tool for engineers to help make their designs better through improved privacy awareness. Therefore, it is important to note that the framework doesn't provide any formal guarantees that IoT systems designed using it will be free of potential privacy problems. However, we believe software engineers will at least be able to apply some privacy guidelines to their design, which they would not do otherwise. Mostly, we wanted to guide individuals and teams who do not have time, or resources to invest in hiring privacy experts. Completely ignoring privacy issues could cost such small teams a lot it long run as they grow. Later re-factoring is always costly in any software development process. Therefore, our guidelines will help entrepreneurial teams, IoT hackers, hobbyists, etc. to embed privacy protecting features into their IoT application designs at the initial stages without consulting privacy experts. While our guidelines cannot replace privacy experts and consultants in the software engineering process, they can help software engineers to reduce the effort and time needed from privacy experts.

The paper is structured as follows: Section 2 discusses common IoT architectures and their characteristics. It also briefly introduces the data life cycle phases and their importance when designing privacy into IoT applications. In Section 3, we present our motivation through three different use cases. We have used these use cases to evaluate the effectiveness and identify the challenges in designing privacy aware IoT applications. We briefly introduce the PbD framework in Section 4. In Section 5, we explain the research methodology and evaluate the effectiveness the PbD framework. We discuss our findings and lessons learned in Section 6. Finally, Section 7 presents the related work and compares our PbD framework with existing approaches. In Section 8, we conclude the paper by discussing future directions for our research.

2. Internet of Things Software Architecture

In this section we provide an overview of IoT software architectures from the perspective of how data moves through a typical IoT application. As illustrated in Figure 1, in IoT applications, data moves from sensing devices to gateway devices to the cloud infrastructure [26]. This is the most common architectural pattern used in IoT application development, which is also called the centralised architecture pattern [30]. However, there are other patterns such as 1) collaborative, 2) connected intranet of Things, and 3) distributed IoT [30]. Even for these other types of architectures, if we consider a single data item, we can observe a data flow analogous to that of the centralised architecture pattern where data moves from edge devices to the cloud through different types of nodes. Therefore, while we use the centralised IoT architectural pattern to explain our PbD framework in this paper, our approach is agnostic the choice of pattern.

Centralised architectures typically consist of three components: 1) IoT devices, 2) Gateway devices, and 3) IoT cloud platforms (Figure 1), each with different computational capabilities. They also have different types of access to energy sources from permanent to solar power to battery power. Each device may also have limitations as to the type of data processing that can be done due to lack of availability of essential knowledge. A typical IoT application would integrate all these different types of devices with different capabilities. It is important to note that different types of privacy protecting measures can be taken on each of these components based on their characteristics.

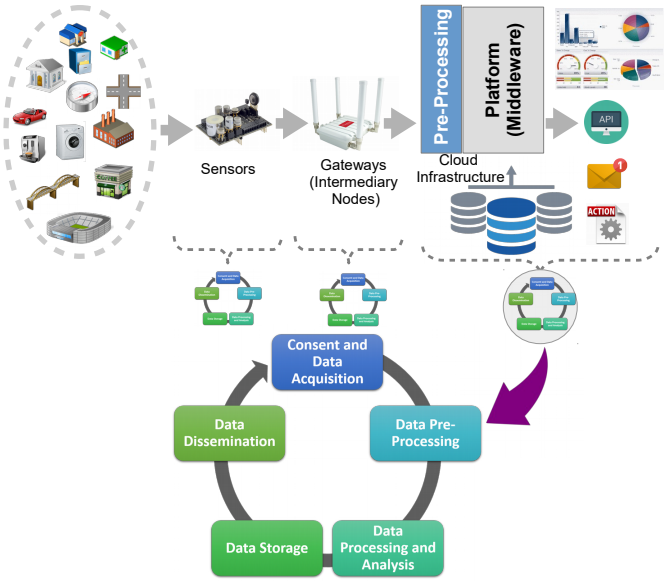

Figure 1: Typical data flow in IoT Applications

We define a five-phase data life cycle that provides a systematic way of thinking about the data flow in an IoT system for the application of our PbD framework. Within each device (also called a node), data moves through the data life cycle phases: Consent and Data Acquisition [CDA], Data Preprocessing [DPP], Data Processing and Analysis [DPA], Data Storage [DS] and Data Dissemination [DD]. The CDA phase comprises routing and data collection activities by a given node. DPP describes any type of processing performed on the raw data to prepare it for another processing procedure. DPA is, broadly, the collection and manipulation of data items to produce meaningful information. DD is the distribution or transmission of data to an external party.

All the data life cycle phases are applicable to all nodes in an IoT application, making it possible for software engineers to put in place appropriate mechanisms to protect user privacy. However, based on the decisions taken by engineers, some data life cycle phases in some nodes may not be utilised. For example, a sensor node may utilise the DPP phase to average temperature data. Then, without using either the DPA or DS phases to analyse or store data (due to hardware and energy constraints) the sensor node may push the averaged data to the gateway node using the DD phase.

3. Example IoT Scenarios

In this section, we present three use case scenarios, which we also use to evaluate the PbD framework as described in Section 5. Each scenario is presented from a problem owner's perspective, where each problem could be solved by developing an IoT application. More importantly, it should be noted that none of these scenarios explicitly highlight privacy requirements or challenges. They are primarily focused on explaining functional requirements at a high level. Later in Section 4, we explain how our PbD framework can be used by software engineers to extract additional information from problem owners, that are crucial to design privacy aware IoT applications.

3.1. Use case 1: Rehabilitation and Recovery

Summary: Robert is a researcher who oversees a number of rehabilitation facilities around the country where patients with physical disabilities are treated and rehabilitated. Robert is interested in collecting and analysing data from sensors worn by patients while they engage in certain activities (e.g., walk using walker, walk using crutches, climb stairs), in order to guide the patients' recovery processes in a more personalised manner. Robert has an application that is capable of analysing patient data and developing personalised rehabilitation plans. The application monitors the progress and alters the rehabilitation plans accordingly. There is a speciality nurse allocated for each patient in order to monitor the recovery progress and provide necessary advice when required.

3.2. Use case 2: Health and Well-being

Summary: Michael works in local government for the department of public health and well-being. He has been asked to develop a plan to improve the public health in his city by improving the infrastructure that supports exercise and recreational activities (e.g., parks and the paths that supports jogging, cycling, and places for bar exercise, etc.). In order to develop an efficient and effective plan, Michael needs to understand the movements of people and several other aspects of their activities. Michael is planning to recruit volunteers in order to gather data using sensors. Michael has an application that is capable of analysing different types of data and recommending possible lifestyle improvements for healthier living. Michael only needs to collect data when the volunteers are within the park premises.

3.3. Use case 3: Amusement Park and Leisure

Summary: *TrueLeisure* is a company that operates different types of franchised entertainment attractions. Their amusement parks are located in the United States, United Kingdom, and Australia. These amusement parks are fully owned and operated by franchisees. However, *TrueLeisure* continuously monitors and assesses service quality attributes and several other aspects at each of the amusement parks. Jane is a data analyst overseeing the quality assessment tasks at *TrueLeisure*. She is responsible for continuously monitoring the service quality attributes. Waiting time is one of the key service quality attributes and is a key determinant of customer satisfaction. Local quality assessment teams continuously measure visitors' waiting time for each ride and attraction within their own amusement park. All visitors use *TrueLeisure*'s theme park mobile app to buy tickets for attractions, further information, tour guide, maps, etc. Jane is interested in the big picture, i.e. she would like to measure the overall waiting time for each ride attraction by combining individual waiting times. Jane will report these measurements to *TrueLeisure* management to guide franchisees on future developments of their theme parks efficiently and effectively.

4. Privacy-by-Design Framework

In each of the example scenarios above, a software engineer would need to perform further analysis to extract explicit privacy requirements that could support the design of privacy enhancing features into the IoT applications that would be developed to deliver the required functionality. In this section we provide an overview of our PbD framework [27] and explain how it could be used to design privacy into IoT applications. We also explain why guidelines are useful to software engineers.

4.1. Why Guidelines (or Heuristics or Check-lists)?

We use the term *guidelines* as our intention is to guide software engineers. In general, a guideline aims to improve or maintain efficiency of a particular

process based on a set of best practices. Guidelines may not be mandatory to follow, but provide recommendations based on experience of dealing with particular problems. Therefore, the term *heuristics* is also an appropriate term for our guidelines. These techniques rely on using readily accessible, though loosely applicable, information to control problem solving in human beings, machines, and abstract issues [24]. Heuristics do not promise to produce perfect or optimal solutions. Finally, the term *check-list* is also appropriate to identify our guidelines. A check-list is a type of informational aid used to reduce failure by compensating for potential limits of human memory and attention. Our guidelines also aim to reduce human errors by reducing knowledge requirements.

Sometimes, guidelines are considered to be a less useful approach due to their inherent characteristics such as: lack of proof (for consistency or correctness), dependence on the subjective judgement of the follower, lack of rigorous scientific methods for extracting guidelines, and so on. Despite such weaknesses, guidelines are being used successfully in many domains. The following examples showcase where guidelines / heuristics / check-lists are successfully used to address different challenges.

- Heuristics based usability design and evaluation is widely used in the human computer interaction domain [22, 21].
- The Information commissioner's office, UK's independent authority set up to uphold information rights in the public interest, use check-lists to guide businesses to prepare themselves for GDPR [1].
- Surgical Safety Check-list developed for the World Health Organization by Dr. Atul Gawande has been able to reduce mortality by 23% and all complications by 40% [12]. Airplane pilots rely upon check-lists to ensure that both routine procedures and emergency responses are handled appropriately [11].

The above usages and successes have given us confidence to integrate guidelines into our PbD framework. The framework combines the guidelines with a method for applying them that avoids the need for individual software engineers to spend time thinking about relevant privacy considerations for their IoT applications themselves. Instead, they can save time and effort by systematically working through the guidelines one by one and checking whether they can apply them. Our node-by-node design methodology also helps manage the complexity of IoT application designs. Guidelines also provide meaningful ways to divide workload among engineers (e.g., each engineer may focus / specialise on addressing a few guidelines) and can be used as a common knowledge base to discuss application designs in teams. Guidelines make the design process comparatively less tiring for engineers as it reduces intensive thinking and knowledge requirements. Guidelines also allow engineers to pause and resume conveniently and

[1] https://ico.org.uk/media/for-organisations/guide-to-data-protection-2-4.pdf

keep track of design changes. We acknowledge that guidelines are not perfect and will need to be reviewed and refined over time. However, evidence suggests that guidelines can help to improve effectiveness and efficiency in a range of situations, and in this paper we demonstrate this in the context of privacy aware IoT application design.

4.2. Overview of the Guidelines

For ease of reference, we present an overview of our privacy guidelines in Table 1. These guidelines are based on Hoepman's [13] privacy strategies, which we determined to be the most appropriate starting point for developing a more detailed set of PbD guidelines for IoT applications. The guidelines were compiled by using the structured-case research method [3], which is often used for building theory in information systems research. A more detailed explanation on each of the guidelines and reasoning behind the extraction of each guideline is presented in [27].

The guidelines allow software engineers to customise them as needed to suit their IoT applications. For example, certain applications will require aggregation of data from different sources to discover particular new knowledge (i.e. new pieces of information). Such approaches are not discouraged as long as data is acquired through proper consent acquisition processes. However, IoT applications, at all times, should take all possible measures to achieve their goals with a minimum amount of data. This means that out of the eight privacy design strategies proposed by Hoepman [13], minimisation is the most important strategy.

In our previous work [27], we identified two major privacy risks, namely, secondary usage and unauthorised access that would arise as consequences of not following the guidelines. Secondary usage refers to the use of collected data for purposes that were not initially consented to by the data owners [18], which can lead to privacy violations. Unauthorised access is when someone breaches the confidentiality of the data during any phase of the data life cycle by gaining access without proper authorisation. The privacy risks of secondary usage and unauthorised access are denoted using the symbols ($\otimes$) and ($\ominus$) respectively to indicate their relevance to each guideline. In Table 1, privacy guidelines are colour coded based on the primary privacy design strategy that they belong to. However, it is important to note that some guidelines may belong to multiple design strategies. For example, **(Guidelines 6)** *minimise data retention period* can primarily be identified as a minimise strategy, but it can also be classified as a hide strategy as it reduces the period for which data is visible.

4.3. Use of Privacy-by-Design Framework

The objective of the proposed PbD framework is to help software engineers to ask the right questions regarding privacy protection when designing IoT applications and their architectures. Our approach integrates privacy guidelines into a framework that includes a method for engineers to start thinking about privacy and incorporate privacy features into IoT application designs. A piece

Table 1: Privacy-by-Design Framework

Guideline	Capability						Minimise	Hide	Separate	Aggregate	Inform	Control	Enforce	Demonstrate	Privacy Risks
	DA	DPP	DPA	DS	DD										
1-Minimise data acquisition	✓	✓					✓	✓							⊗ ⊖
2-Minimise number of data sources	✓						✓								⊗ ⊖
3-Minimise raw data intake	✓	✓					✓			✓					⊗ ⊖
4-Minimise knowledge discovery			✓				✓								⊗
5-Minimise data storage				✓			✓								⊗ ⊖
6-Minimise data retention period				✓			✓	✓							⊗ ⊖
7-Hidden data routing	✓				✓			✓							⊖
8-Data anonymisation	✓	✓	✓		✓			✓							⊗ ⊖
9-Encrypted data communication	✓				✓			✓							⊖
10-Encrypted data processing		✓	✓					✓							⊖
11-Encrypted data storage								✓							⊖
12-Reduce data granularity	✓	✓	✓		✓		✓	✓							⊗
13-Query answering					✓		✓	✓							⊗
14-Repeated query blocking					✓		✓	✓							⊗
15-Distributed data processing			✓						✓						⊗ ⊖
16-Distributed data storage				✓					✓						⊗ ⊖
17-Knowledge discovery based aggregation	✓	✓	✓	✓	✓					✓					⊗
18-Geography based aggregation	✓	✓	✓	✓	✓					✓					⊗
19-Chain aggregation	✓	✓	✓	✓	✓					✓					⊗
20-Time-Period based aggregation	✓	✓	✓	✓	✓					✓					⊗
21-Category based aggregation	✓	✓	✓	✓	✓					✓					⊗
22-Information Disclosure	✓	✓	✓	✓	✓						✓			✓	⊗
23-Control	✓	✓	✓	✓	✓							✓	✓		⊗ ⊖
24-Logging	✓	✓	✓	✓	✓									✓	⊗ ⊖
25-Auditing														✓	
26-Open Source														✓	
27-Data Flow Diagrams														✓	
28-Certification														✓	
29-Standardisation														✓	
30-Compliance														✓	⊗ ⊖

Risk Types: Secondary Usage (⊗), Unauthorised Access (⊖)

of software is designed to solve a problem. Sometimes, a problem may be identified by a person who is affected by the problem (e.g., Robert, Michael or Jane in our motivating scenarios). At other times, a third party company may identify a generic problem that affects many other people (e.g., Enterprise resource planning solutions). This type of software engineering is common in the IoT domain as well. Some IoT solutions are generic middleware platforms that can be used to build end to end applications. Others are complete IoT applications that aim to solve a specific problem [26, 25].

However, problem owners mainly focus on the requirements that would help to solve their problem [1], ignoring privacy considerations. Therefore, privacy requirements are largely overlooked when designing software architectures for IoT applications. The PbD framework allows both problem owners and software engineers to sit together and discuss the problem and incorporate privacy protecting measures into IoT application designs.

In section 3, we presented three use case scenarios. For each scenario, we have a problem owner's expectation and a brief set of requirements. There is no explicit reference to privacy protecting measures. We assume, additional information can only be gathered through questioning the problem owners and domain experts. In the studies reported later in this paper, we simulated such discussions between the problem owners (represented by ourselves, the researchers) and the software engineers (represented by the study participants). Our hypothesis was that the PbD framework will help software engineers ask questions from both problem owners and domain experts in order to extract detailed requirements that could be used to design privacy features into IoT applications.

To illustrate how this might work in practice, let us revisit the scenario presented in section 3.1 and use our PbD framework to extract privacy requirements for designing a privacy-aware IoT application.

Guideline 1 leads software engineers to ask the question: what types and quantities of data are required to achieve the Robert's objective? In our scenario the problem owner responds as follows:

Robert collects data using wearable sensor kits. The collected data types are pulse, oxygen in blood (SPO2), airflow (breathing), body temperature, electrocardiogram (ECG), glucometer, galvanic skin response (GSR-sweating), blood pressure (sphygmomanometer), patient activity (accelerometer) and muscle / eletromyography sensor (EMG). The accelerometer data is used to derive patient activity. In addition to the sensor data, weather information such as temperature, humidity are also important for the Robert's research. Patients' mobile phones GPS sensors and weather APIs are used to collect such information. The data collection sampling rate is expected to be 30 seconds. Data is only required to be collected when patients are performing either one of the monitored activities (i.e. walking with walker or crutches, or climbing stairs).

Based on this information the software engineer can decide not to acquire any other types of data and also design appropriate sampling rate controls into the application. This will have the effect of minimising data acquisition and reducing the risk of both secondary usage and unauthorised access to private data.

In a similar fashion, guidelines 3, 5, 20 and 21 would lead a software engineer to ask questions such as: what type of data is required in raw formats and what type of information can be aggregated in order to reduce privacy risks?. As a result, the following information may be gathered.

Robert requires oxygen in blood (SPO2), airflow (breathing), body temperature data types in raw formats which need to be accurate. The data collection sampling rate is expected to be five seconds. In contrast, other data items can be aggregated into averaged values (e.g., aggregated over two minutes).

Similar guidelines based questioning can be used to extract privacy requirements which the software engineers can use to systematically design privacy into the IoT application. Due to space limitations, we don't detail all the questions that could be asked in relation to the scenario. Instead, below we provide the information that could be acquired using our PbD approach by annotating a detailed description of the scenario with references to the relevant PbD guidelines at the end of each statement.

*The sensor kit is expected to push data to the patient's mobile phone using Bluetooth. The mobile phone pushes data to the rehabilitation centre's local server using Wi-Fi. The local server pushes data to the cloud IoT platform. Patients come to the rehabilitation centre 3 days a week in order to perform the tasks assigned to them. Over a further 3 days they perform the tasks in their homes. The smart phone is expected to push data to the local server at the end of each day (**Guideline 6**). However, if the patients perform their tasks at home, data need to be stored on the mobile until they next visit the rehabilitation centre (**Guideline 6**). The speciality nurses monitor the progress and advise the patients on weekly basis. The speciality nurses' responsibility is to make sure that the patients are performing the tasks as assigned by the recommendation system and assists patients if they have any difficulties in following the assigned tasks and schedules. Robert is required to analyse data every six months in order to understand the how to improve the rehabilitation processes in a personalized manner (**Guideline 6**). For long term data analysis purposes, Robert's application stores data after averaging over five minutes (**Guideline 6**).*

*Robert's application requires averages over five minutes when patients are performing their tasks (**Guideline 20**). Patient data can be anonymized (**Guideline 8**). Data storage in both mobile device, local server and Robert's cloud server should store data in encrypted form (**Guideline 11**). End-to-end encryption can be used to secure the data communication (**Guideline 9**). Robert does not require the exact locations where patients may have performed the activities. The requirement is to acquire the weather parameters such as temperature, humidity, etc. Therefore, location data can be abstracted without affecting the accuracy of the data (**Guideline 12**). In this IoT application, data processing and storage happens in three different nodes, namely, 1) patient phone, 2) local server, and 3) Robert's cloud server (**Guideline 15 and 16**).*

The above example illustrates how the PbD guidelines could be used to extract additional information regarding a use case which enables software engineers to design appropriate privacy enhancing features into their IoT applica-

Table 2: Relevant Privacy Requirements for Each Use Case Scenario

Guideline (↓) Use Case Number (→)	1	2	3
1-Minimise data acquisition	✓	✓	✓
2-Minimise number of data sources	–	✓	–
3-Minimise raw data intake	✓	✓	✓
5-Minimise data storage	✓	✓	✓
6-Minimise data retention period	✓	✓	✓
7-Hidden data routing	✓	✓	✓
8-Data anonymisation	✓	✓	✓
9-Encrypted data communication	✓	✓	✓
11-Encrypted data storage	✓	✓	✓
12-Reduce data granularity	✓	✓	✓
15-Distributed data processing	✓	✓	✓
16-Distributed data storage	✓	✓	✓
18-Geography based aggregation	–	–	✓
20-Time-Period based aggregation	✓	✓	✓
21-Category based aggregation	✓	✓	✓
	13	14	14

tions. In order to evaluate the effectiveness of our PbD framework, we developed similar detailed requirement descriptions for each of the use case scenarios, which we have omitted here due to space limitations. It is important to note that not all privacy guidelines are relevant to all IoT applications. In Table 2, we summarise which privacy guidelines are relevant to each scenario.

5. Evaluation

This section explains how we evaluated our PbD framework, together with our research methodology. Specifically, our evaluation is based on the following two studies:

1. **Study 1 (Primary)**: [Interview based] This was our primary study in which we tested our main hypothesis: *'Can the proposed PbD framework guide software engineers with varied levels of experience to design IoT applications that are more privacy-aware than they would do otherwise?'* Additionally, we explored engineers' perception of each guideline, their usefulness, and applicability in different IoT use case scenarios - collectively referring to this as the engineers' *privacy mindset*. The study was administered by a researcher and focused on both quantitative (for hypothesis testing) and qualitative data.
2. **Study 2 (Secondary)**: [Online activity based] This was a self administered online study. In this study, we explored the engineers' privacy mindset with respect to each guideline. In contrast to Study 1, here we used an anonymous, informal, and relaxed methodology using a self administered online activity that could be completed over a 3-day period. We

used this study to strengthen our findings from Study 1 as well as to reach theoretical saturation[2]. In this study, we mainly focused on qualitative data (though we present some quantitative aspects).

For each study, we first explain the aims of the study followed by a description of the participant recruitment strategy and sample size. Finally we describe the procedures followed at each step of the study. In adopting this approach, we were partially inspired by the evaluation strategies used by comparable techniques, particularly the evaluation methodology used for LINDDUN [7], including adopting a use case based evaluation technique.

5.1. Study 1 (Primary) - Interview-based

5.1.1. Purpose

The purpose of this study is to explore how our PbD framework can help software engineers to design privacy-aware IoT applications. Through user studies, using quantitative and qualitative data analysis, we aimed to answer following three questions that explore the effectiveness of the proposed PbD framework. We discuss these questions later in this section.

- Can the proposed PbD framework guide less experienced (novice) software engineers to design IoT applications that are more privacy-aware than the applications they would design without the guidelines?
- Can the proposed PbD framework guide more experienced (expert) software engineers to design IoT applications that are more privacy-aware than the applications they would design without the guidelines?
- Out of novice and expert software engineers, who would benefit most from the proposed PbD framework? or in other words, does the level of software engineering expertise matter when it comes to incorporating privacy protection features into IoT application designs?

In the first two questions above, we consider the design of an IoT application to be more privacy-aware if the designer considers a greater number of privacy concerns to incorporate appropriate privacy protecting features. We measure this in terms of the number of privacy guidelines considered by the study participants when designing the example IoT applications.

5.1.2. Recruitment and Remuneration

In total, we recruited 10 participants for the study of which five were novice software engineers and five were expert software engineers. A participant was classified as a novice if they had less than three years of experience (full-time) in a software engineering role (design or development). Participants with more

[2]Theoretical saturation is the phase of qualitative data analysis in which the researcher has continued sampling and analysing data until no (or very minimal) new data appear [17]

than three years of experience (design or development), were considered to be experts. We adopted an opportunistic sampling technique and participants were recruited from the staff and student populations across two universities in the United Kingdom. No criteria other than software engineering experience was considered when recruiting participants. We collected demographic information such as age, highest education qualification, and the number of years in a software engineering role. Each participant was compensated with shopping vouchers valued at GBP 20. There were no failure criteria as long as the participant attend the data collection session of the study. The study design was reviewed and approved by our institution's Human Research Ethics Committee. Table 3 summarises the demographic information about the participants. We have labelled them E1-E5 (Expert) and N1-N5 (Novice) and consider them to be independent cases for the purposes of our qualitative analysis process.

5.1.3. Procedure

All the data collection sessions were carried out as 1-to-1 lab-based observational studies. The principal investigator (PI) acted as the facilitator as well as the observer during each of the sessions. The duration of each session was 1.5 hours. At the beginning of the each session, participants were given the consent form to sign off and brief demographic information was collected. We audio recorded all the discussions between the participants and the PI for qualitative analysis purposes. Next, participants were given an instruction sheet, as shown in Figure 2, that comprised a set of example notations that could be used to illustrate the design of the IoT applications. Participants were reassured that adherence to the notation was not essential.

We divided the rest of the study into three rounds, first with no guidance to consider privacy or reference to the PbD framework (Round 1), then with a prompt to consider privacy requirements for the use cases but no reference to the PbD framework (Round 2), and finally using the PbD framework (Round

Table 3: Demographics of Study 1 (Primary Study)

ID	Age	Highest Qualification (ICT)	Years of Experience	Area of Experience
E1 (Male)	20-29	MSc	4 (Expert)	Desktop, Mobile, Web
E2 (Female)	30-39	PG(Diploma)	8 (Expert)	Mobile, web, system integration
E3 (Female)	30-39	MSc	8 (Expert)	Embedded, Textile Design, wearable
E4 (Male)	40-49	BSc	10 (Expert)	Data Science
E5 (Male)	20-29	BSc	6.5 (Expert)	Desktop, Mobile, Web
N1 (Male)	30-39	PhD	3 (Novice)	Signal Processing
N2 (Male)	30-39	MSc	2.5 (Novice)	Desktop
N3 (Male)	20-29	BSc	3 (Novice)	Desktop
N4 (Male)	30-39	MSc	1 (Novice)	Desktop
N5 (Male)	30-39	MSc	3 (Novice)	Web

3). However, this segmentation was only used to structure the discussions and observations and none of the rounds were formally acknowledged or identified during the sessions.

Round 1 (NoPrivacy) - *IoT application design without any guidance to consider privacy or reference to the PbD guidelines*: It is important to note that we informed the participants that this is an IoT application design study, without making any reference to privacy. This was done with the expectation that participants would be unbiased and follow their natural process for designing an IoT application. We gave them separate A4 sheets to draw their IoT application designs with respect to each use case. They were briefed about the notations they could use, but we did not restrict them to any particular notation as long as their designs were understandable and clearly annotated.

The participants were asked to design IoT applications to satisfy the requirements of each the scenarios presented in Section 3. Initially the participants worked from the summary descriptions provided in this paper but the PI was prepared to provide more detailed information, similar to that presented in Section 4.3 if the participant explicitly asked any related questions. We designed the study to simulate a conversation between a software engineer and a problem

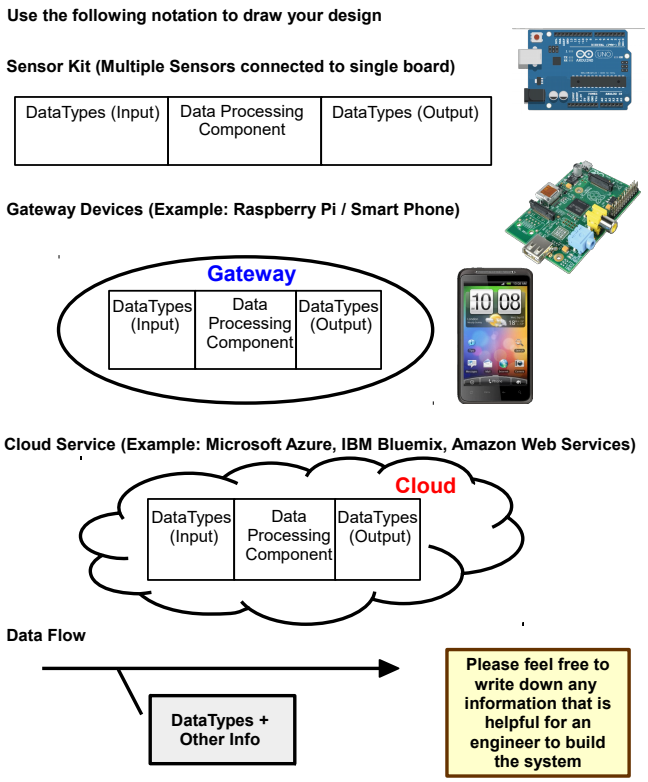

Figure 2: Notations to be used in IoT application Design

owner, where the engineer is trying to elaborate the requirements and design the architecture of the IoT application.

We encouraged participants to ask as many question as possible about the case studies and application requirements. This means that participants could have asked any question regarding privacy requirements if they wanted to. Some of the commonly asked questions are discussed later in this paper. We gave them 50 minutes to complete the IoT application designs for the three use cases provided. However, the time limit was only a guide to the participants and was not enforced. The actual time taken for each study varied based on the time taken by the participants on each phase. So the actual total time varied between 1 hour and 15 minutes to 2 hours. We always allowed each participant to naturally progress through their designs without rushing them through each phase. After the designs were completed, we asked the participants to explain their designs and briefly justify their design decisions.

Round 2 (WithPrivacy)- *IoT application design with guidance to consider privacy but without privacy guidelines*:Next, we gave participants a ten minute introduction on privacy. In order to achieve consistency, accuracy, and a well recognised description of privacy and related challenges, we selected two videos[3] [4] from YouTube produced and published by *Privacy International* (www.privacyinternational.org). The objective of showing these videos to each participant was to provoke them to think about privacy and help them to recall their past experiences and knowledge of dealing with privacy issues. This was intended to help them with the next task. It is important to note that we did not provide any additional material on privacy at this stage.

Next, we asked the participants to refine their previous IoT application designs further to protect user privacy. Similar to the previous round, questions were welcomed. We gave the participants 20 minutes to refine the IoT application designs for the three use cases provided. For Round 2, they wrote in a different colour to Round 1, which enabled us to distinguish the design activities from each round clearly. After the revisions were made, we asked the participants to explain their revised designs and how they improved privacy protection.

Round 3 (WithPbDGuidelines) - *IoT applications design with privacy guidelines*: Finally, we gave participants an introduction to the PbD guidelines and how to use them. We asked the participants to refine their previous IoT application designs to protect user privacy. Similar to the previous round, questions were welcomed. We gave the participants 20 minutes to enhance the privacy features of their IoT application designs for the three use cases provided. After the revisions were made, we asked the participants to explain their revised designs and how they improve privacy protection. Once completed, we collected the IoT application designs produced by the participant. Some sample application designs produced by participants are presented in Figure 3.

[3]What Is Privacy? (youtube.com/watch?v=zsboDBMq6vo)

[4]Big Data (youtube.com/watch?v=HOoKhnvoYkU)

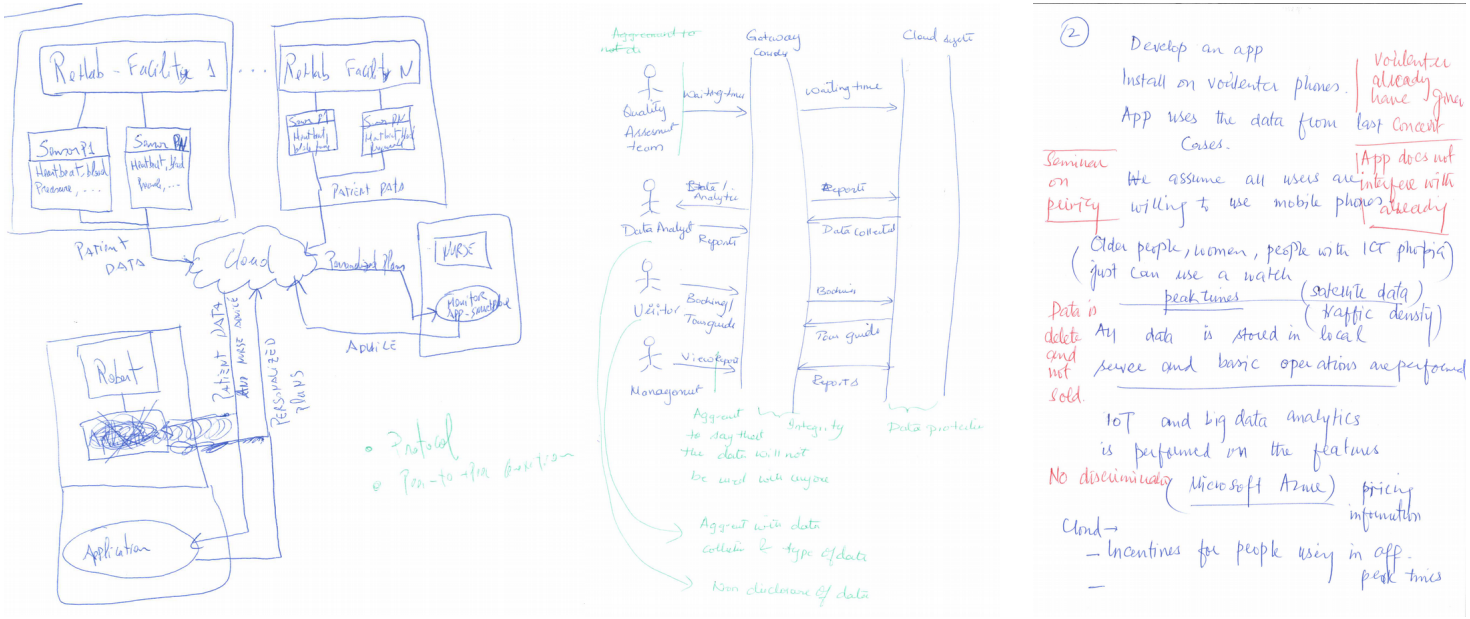

Figure 3: Sample IoT application designs that illustrate a variety of approaches used by participants to express their high-level designs. In addition to block diagram notations based on the examples we provided, participants used sequence diagrams, pictorial diagrams and detailed text descriptions as illustrated above.

5.2. Study 2 (Secondary) - Online Activity-based

5.2.1. Purpose

Study 1 was conducted by a researcher using an interview-based approach. Therefore, participants may have been compelled to think and perform harder during the study. On the other hand, at times we failed to convince the engineers to apply certain guidelines into a given IoT application scenario. In real world situations, these PbD guidelines would need to be used by engineers without supervision (or assistance). By taking these factors into consideration, we designed a second study aimed at exploring the engineers' mindset towards the PbD guidelines. More specifically, we explored what software engineers think about each guideline, their reasoning and decision making process when applying them. It is important to note that the data gathered in Study 2 addresses the same question as Study 1 (Round 3), albeit in a different context. We used Study 2 to strengthen the findings of Study 1 as well as to reach theoretical saturation [17] and we will compare these results in Section 6.

5.2.2. Recruitment

In total, we recruited 17 participants, one of whom dropped out, giving us a final set of 16 participants. This survey, which was conducted at a French University with participants who were Masters students and were recruited using a convenience sampling approach. No compensation was given to the participants. The study involved completing 32 IoT use case scenarios. Based on the lessons we learned from Study 1, we did not consider the level of experience to be a relevant factor in this study. The demographic summary for the participants is presented in Figure 4.

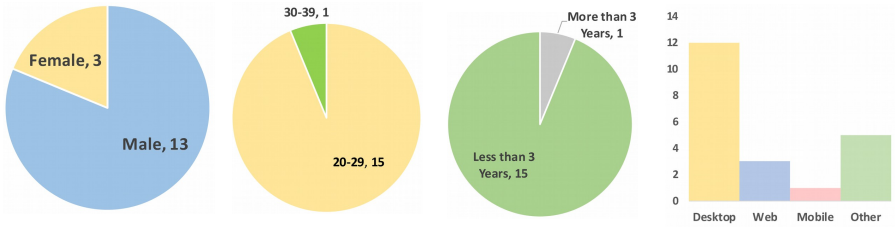

Figure 4: Demographics of Study 2 (Secondary Study)

5.2.3. Procedure

This study was organised using two online surveys. Each participant was given three days to complete the activity. As in Study 1, we used the three case studies presented in Section 3. We formulated the each survey into two logical rounds (in contrast, to the three rounds in Study 1), *1) without privacy guidelines*, and *2) with privacy guidelines*:

Round 1: A use case scenario is presented to each participant. Then, we asked the question *"What kind of privacy protecting measures might you incorporate into the IoT application design?"*. We also recommended the participants

to sketch a data flow diagram saying *"Even though it is not required, it might be useful for you to sketch a data flow diagram to understand how you might want design the IoT application"*.

Round 2: In this round, we presented different PbD guidelines, one by one, and asked the participants to answer appropriately. (*"Please read the above privacy guideline. Do you think this guideline can be applied to the IoT application design? If 'Yes'; please briefly explain how you might apply this guideline. If 'No': Please explain why this guideline cannot be applied").*

6. Findings, Discussion and Lessons Learned

In this work, we followed the multi-method - multi-strand method [37]. More specifically, we used two data collection methods (i.e., interviews and online activities) and collected multiple types of data (i.e., IoT application designs [*drawings*]), participants views [*audio*], participants ability to identify privacy preserving measures [*numeric*]). In this section, we first analyse and discuss the results quantitatively. Our aim is to address the three questions presented earlier in Section 5.1 with the help of data collected through Study 1. Later, we discuss the results of both Study 1 and 2 qualitatively in order to understand software engineers' approach towards designing privacy-aware IoT applications.

6.1. Quantitative Analysis (Exploring Effectiveness)

As shown in Table 2, in Study 1 we expected each participant to identify a maximum of 41 privacy protecting measures (Use-case 1: 12 measures, Use-case 2: 14 measures, Use-case 3: 14 measures). The participants may identify these privacy measures either using their experience, common sense, or using the PbD guidelines. In total, we collected 410 data points (41 measures x 10 participants). We present an overview of the data gathered using two heat-maps in Figure 5 where the results for novice and expert software engineers are presented separately.

The heat-maps clearly show that both novice and expert software engineers were able to identify a greater number of privacy protecting measures by using the PbD guidelines than they would do otherwise. In Figure 6, we illustrate how the mean of the *'number of privacy measures'* identified changes at different privacy knowledge levels, for novice and experts software engineers. The average number of privacy measures identified, in Round 1, by novices is 0.2 and experts is 2.2. Similarly, the average number of privacy measures identified, in Round 2, by novices is 6.6 and experts is 6.8. Further, the average number of privacy measures identified, in Round 3, by novices is 32.6 and experts is 30.4.

Next, we ran statistical tests (i.e., ANOVA[5]) and found out that there is a significant difference between the number of privacy measures identified with and without the PbD guidelines (within=PrivacyKnowledge (ANOVA p =

[5]statistics.laerd.com/statistical-guides/one-way-anova-statistical-guide.php

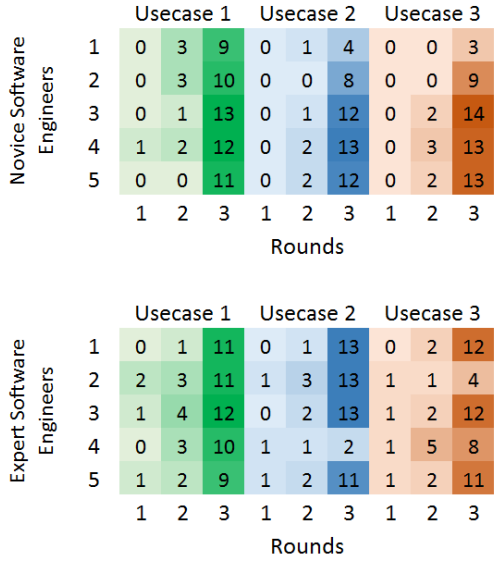

Figure 5: The three use-cases are marked using three separate colours. The x-axis denotes how many privacy protecting measures have been identified in each round (the darkness of the sharing is proportional to the number of privacy requirements identified). The y-axis denotes the participant ID.

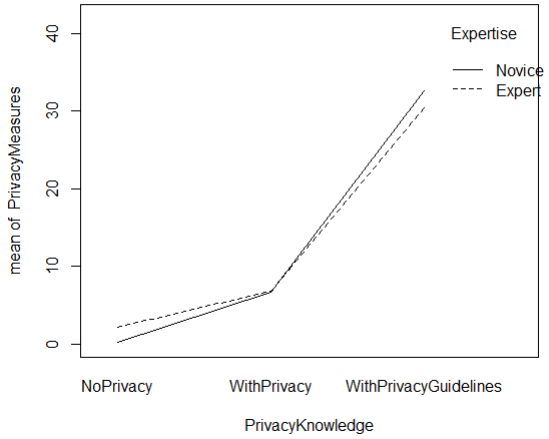

Figure 6: Number of privacy measures identified in each round

2.099781e-09; $p < 0.05$)). Further, our results show that the expertise of the software engineers (novice vs. expert) has no significant effect on the identification of privacy protecting measures (between=Expertise (ANOVA p = 6.897806e-01; $p < 0.05$)) .

Figure 7 illustrates which privacy guidelines have been identified in each round by the participants. It is important to note that PbD guideline 2 and 18

were only relevant in one of the use case scenarios which explains its unusually low identification rate in Figure 8. To avoid any confusion, we have presented the x-axis of the Figure 8 as a percentage. Comparatively, more participants have identified PbD Guideline 3 (Minimise raw data intake) and 20 (Time period based aggregation) in Round 1. However, our discussions revealed that participants integrated these features into their designs to meet functional requirements of the scenarios rather than due to a consideration of privacy. In Round 2, after we explicitly asked them to improve the privacy awareness of their IoT application designs, participants primarily identified Guideline 8 (data anonymisation), Guideline 9 (encrypted data communication), and Guideline 11 (encrypted data storage). In Round 3, there was no noticeable difference in the guidelines identified by the participants.

Results from both Study 1 (Figure 7) and Study 2 (Figure 8) are comparable, showing that participants mostly understand and agree with the usage of encryption (communication and storage) and data minimisation very well. However, we observe a higher refusal / disagreement rate in Study 2. We discuss this phenomenon further in Section 6.2.

In total, we expected participants to identify a maximum of 410 privacy preserving measures that they could take in order to improve the privacy awareness of the three given IoT application scenarios. They identified 308 privacy preserving measures with the help of the PbD guidelines, giving a success rate of 75.12%. As shown in Figure 6, this result is significantly better than *'without PbD guidelines'*. Based on our discussions with the participants, we identified two main reasons why they sometimes failed to apply a given guideline to their designs: 1) the IoT application designs eliminates the need to apply certain privacy preserving measures; and 2) the lack of time. The former reason arises because PbD guidelines can only be applied in certain application design con-

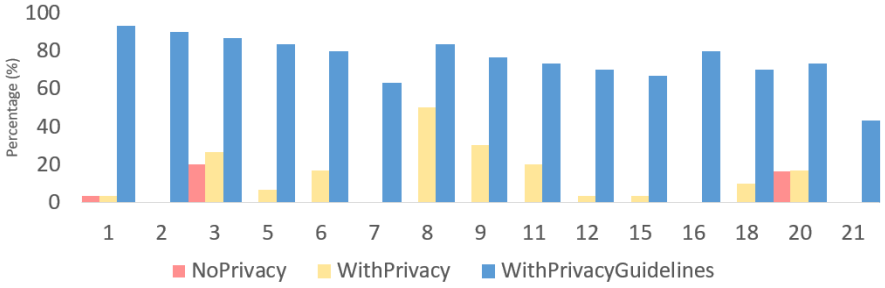

Figure 7: Study 1 - Privacy guidelines identified in each round: the x-axis denotes privacy guidelines by number and each colour represents the three rounds. The y-axis denotes the frequency with which participants identified a given privacy guideline. Legend for both Figure 7 and Figure 8: 1-Minimise data acquisition, 2-Minimise number of data sources, 3-Minimise raw data intake, 5-Minimise data storage, 6-Minimise data retention period, 7-Hidden data routing, 8-Data anonymisation, 9-Encrypted data communication, 11-Encrypted data storage, 12-Reduce data granularity, 15-Distributed data processing, 16-Distributed data storage, 18-Geography based aggregation, 20-Time-Period based aggregation, 21-Category based aggregation.

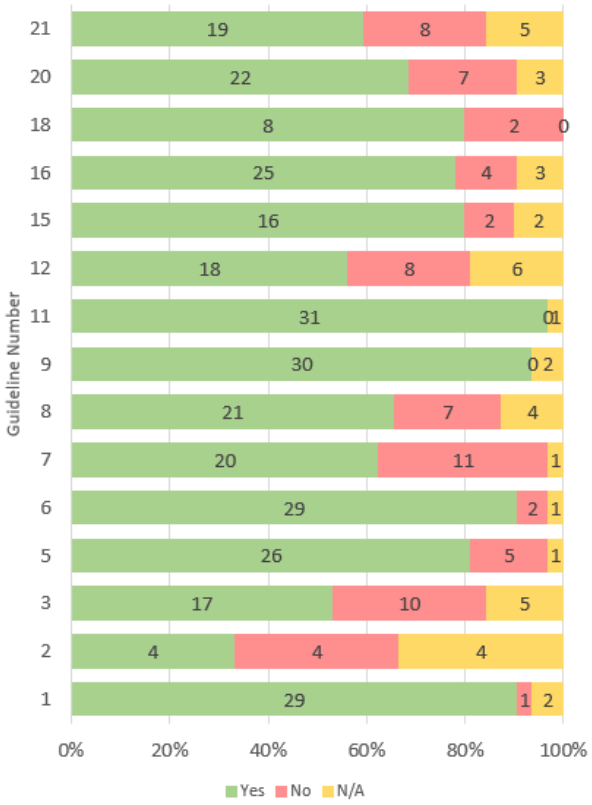

Figure 8: Study 2 - Participants' view on whether a given guideline can be applied or not to the given IoT use case scenario. Legend: Yes = participant agrees that a given guideline can be applied; No = participant refuses to apply a given guideline; N/A = participant did not clearly specify whether the guideline is applicable or not.

texts. Some participants designed their IoT applications such that certain PbD guidelines were not relevant. We discuss one such example in the next section.

6.2. Qualitative Analysis and Lessons Learned

We followed Miles framework [20] to conduct the qualitative analysis. Further, for data reduction phase, we use Richards three tier coding technique (i.e., descriptive coding, topic coding, and analytic coding) [29]. This yielded three broad themes relating to, (1) ***methodological*** lessons for the PbD framework for IoT, (2) lessons about the ***cognitive*** behaviour of engineers when designing privacy into IoT applications, and (3) ***practical*** lessons relating to PbD for IoT applications (4). We provide further details of the lessons learned through the qualitative analysis below.

Table 4: Qualitative analysis: Themes and lessons

Themes	Lessons
Methodological	Ambiguity in use case description; Duration of the tasks; Task authenticity
Cognitive	Privacy mindset; Exploratory cues; Knowledge gaps; Adoption barriers; Guideline descriptions; Supporting consistency; Unconscious bias; Over analysis
Practical	Contextual advice; Limitations of informed consent; Responsible engineering; Alternative technologies; Building on adequate security

6.2.1. Methodological Lessons

By observing our study participants engagement with the PbD framework, we identified lessons relating to the method for applying our approach.

First, we intentionally kept the use cases descriptions used in the study brief, expecting that participants would ask questions to clarify any ambiguities. From the participants perspective, this provided an authentic experience of discussing requirements with a stakeholder as part of producing a design. However, none of these discussions developed into privacy requirements gathering. Instead, participants questions focussed on functional and technological requirements. Indeed, only one (out of 10) participants in Study 1 explicitly discussed privacy requirements during one of the Round 1 designs. For example, when formulating a design for Use Case 2, participant [E1] said *Thinking about issues as privacy, for example, I would just be interested to know how many are there and not who is there. By that I could, for example, use the signal of the mobile phone and identify how many mobile phones are there.* This demonstrates the importance of having PbD frameworks for IoT application design that will prompt engineers to explore privacy issues due to ***ambiguity in use case descriptions***.

Another methodological lesson stems from the ***duration of the tasks*** involved in applying the PbD framework. The total duration of the activity in Study 1 was about 1.5 hrs. Going through privacy guidelines and deciding when, whether, or how to apply them is a significant and tiring task, especially when the number of guidelines is significant. However, if we try to reduce the number of guidelines, this will increase the abstractness and ambiguity of each guideline (e.g., Ann Cavoukian [5]). This aspect highlights the need for tools to support the application of the PbD framework.

Finally, from the methodological perspective we noted the impact of ***task authenticity*** on the participants behaviour. In particular we observed more examples of engineers reluctance to adopt privacy guidelines in Study 1 (Round 3) when compared to Study 2 (Round 3). We attribute this to the self-administered, unsupervised context of Study 2 where participants had independence and flexibility over how that explored and applied the guidelines. We note that these features made Study 2 a more authentic experience of real-world design contexts where software engineers have to use PbD guidelines by themselves.

6.2.2. Cognitive Lessons

The second theme we identified from the qualitative analysis relates to the behaviour and thought processes of the engineers as the completed the tasks.

A key lesson in this theme is the need for engineers to develop a ***privacy mindset***. This is a way of thinking that consistently considers privacy as part of the design process. The absence of a privacy mindset is illustrated by the example of participant [N1] who recognised the importance of anonymising and deleting the data with regards to Scenario 1 in Round 2, but was reluctant and refused to apply the same ideas to Scenario 3. Specifically, they said *I mean I can see a whole bunch of scenarios where they would want to pitch different kinds of deals to these individuals. Thats why Im saying its very unlikely that they would adopt any sort of privacy enhancement measure, to get rid of or de-anonymise that data.'* The absence of a privacy mindset is also illustrated by participants superficial consideration of privacy. For example, participant [E4] was not particularly interested in thinking about privacy from certain aspects such data minimisation saying that *So, that information would, in theory, it might be possible to infer from the raw data, but in practice that could be quite tricky (Laughter).* There were also examples of participants treating privacy as a secondary objective, focusing on collecting as much data as possible (e.g., Participants [E2] said *As a developer, you just want all of the data*). The PbD framework presented here supports engineers in developing a privacy mindset.

One way in which the guidelines support engineers was in providing ***exploratory cues***, prompting them to draw on wider expertise to address privacy issues. Privacy guidelines can effectively educate and inform intelligent, but non specialist engineers and designers. For example, participant [E2] mentioned that *The distributed data processing, I had not thought about at all to be honest I do not think but yes, I think it could definitely apply to all of these in some way. I am not sure how because I do not work in networks, or do this kind of stuff but I think that it would be good.'*

Another lesson was that ***knowledge gaps*** in engineers understanding of how different technologies can enhance privacy. For example, participant [E1] of Study 1 mentioned that *I guess it is not necessary to encrypt and anonymise the data.'* They did not understand that encryption and anonymisation techniques are designed for different purposes, providing two lines of defence. We also observed ***adoption barriers***, when discussing the application of the guidelines. For example, in Study 1, participant [N5] refused to apply categories based aggregation guidelines, saying *Yes, I understood, but I dont think that we need the categories based aggregation.*

We also noted the influence of ***unconscious bias*** stemming from the prior expertise of the engineers. For example, in Study 1, participant [N1] implicitly thought about data minimisation from a networking perspective: *Is it sort of a very low-bitrate data that you can collect on the cloud and analyse later? Im wondering if you need to do any data processing at all?* This shows how engineers may implicitly apply certain guidelines without thinking about privacy, instead thinking about challenges in their own expert areas.

Some participants struggled to maintain a consistent approach across the different scenarios. For example, participant [E1] suggested using secure protocols for communication with regards to Use Case 1 even before seeing our privacy guidelines. However, they did not suggest using secure protocols for Use Cases 2 & 3. This illustrates how the PbD guidelines can ***support consistency*** in designing privacy into IoT applications.

We noticed that using guidelines could be tricky and engineers were prone to ***over analysis*** when applying them. For example, participant [N5] mentioned that *Distributed data processing, I did not think about this before reading the guidelines. For scenarios two and three, we can distribute the data for processing. We send them to different clouds.'* Even though distributed processing is applicable in the scenario, attempting to employ multi-cloud processing as a way to apply distributed processing could lead to unnecessary complexity and higher costs with little contribution to privacy protection. This illustrates the importance of assessing each context carefully before applying a particular guideline.

Finally, we noted that the ***guideline descriptions*** caused some challenges to engineers. For example, participant [E1] of Study 1 asked *Whats the difference between the reduced data granularity and the minimise data acquisition.'* We observed similar remarks in Study 2 as well where participants mentioned that they do not understand certain guidelines or how they can be helpful. Enhancing the guidelines descriptions to provide an example should address these types of issue.

6.2.3. Practical Lessons

The final theme for the lessons identified from our analysis relates to practical aspects of integrating privacy into the design of IoT applications.

We noted that engineers in Study 1 often found it difficult to apply the guidelines to the particular context of the use case. For example, participant [E1] said that *In this case, he needs to know which one person it is. Its important because the personal is a person then I cant anonymise or blow it. Yes. Because this one is really a personal thing, so I think the main problem is the cloud.* However, this is not correct. In Use Case 1, personal information can be replaced by an identifier (for example, without using the real name. However, in this particular instance, [E1] concluded that personal data has to be retained. In a different example from Study 1, participant [N4] refused to apply the minimise raw data intake guideline saying that *I think it was not considered in scenario three, where I said that we will be sending the video feed to the Cloud. That can actually give the information regarding a particular user at that particular place.* They only realised that sending the entire video was unnecessary when prompted by the researcher. Addressing these kinds of problems requires ***contextual advice***, which can be formulated as patterns and integrated into automated design support tools.

Another lesson was the need for engineers to understand the **limitations of informed consent**. We noted several examples where participants used consent forms as a mechanism to justify data collection, processing and storage. For

example, participant [E2] mentioned that *Assuming that all the patients were part of the trial that the researcher is doing, and had already signed up to allowing the data to be tracked.* Further, they mentioned that *These were volunteers, so, under the assumption that theyve been signed up and made fully aware that this is going to track their movements.* However, such a data collection approach is not allowed under the new GDPR regulations [8] where all the data collection and retention activities need to be justified. We made similar observations in Study 2 as well. One way to address this issue is to use the PbD framework to develop the engineers privacy mindset.

The studies also highlighted that participants understood the need for ***responsible engineering*** approaches to embed privacy into IoT application design. Indeed, many participants followed the guidelines and successfully improved their designs, but also claimed that they thought about privacy considerations before we showed them the guidelines, even though their designs did not show any evidence of this. We observed two different types of responses: (1) *revisionist* answers where the participant says that they have thought about a certain guideline, but there is no evidence to support this (e.g., *So, I think I did consider the minimising the data that has been recorded* [E2]); and (2) *reluctant acknowledgement* that they havent thought about it (e.g., *It is tricky actually because when you are thinking about stuff you are like I kind of understand it, but I was not really thinking that at the time.* [E3]).

It is also important for engineers to consider the privacy implications of ***alternative technologies*** when designing IoT applications. In relation to Scenario 2 (Section 3.2), one participant [E4] used stationary sensors that do not capture any personally identifiable information to collect the necessary data. This shows how privacy risks can be reduced by selecting certain types of sensing technologies provided they are appropriate for the IoT application being developed.

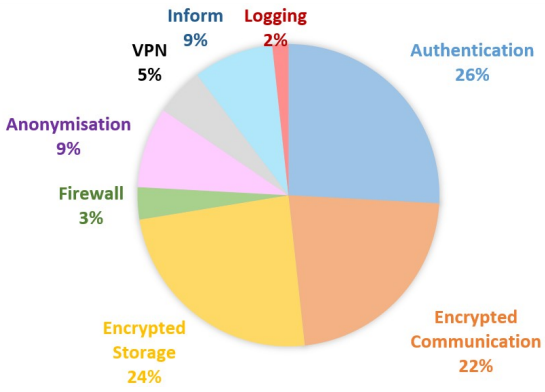

Figure 9: Privacy protection measures identified in Study 2 (Round 1 - Before seeing the PbD Guidelines)

As shown in Figure 9, the most common privacy protection measures identified are authentication and encryption. However, in our PbD framework, we considered these to be security measures rather than privacy protection measures. Study 2 (Round 1) also highlighted the same issue. This demonstrates that engineers know that ***adequate security*** is important for privacy, but that they dont necessarily understand that such security measures are not sufficient to ensure that privacy is maintained.

6.3. Limitations

Although all the participants were able to understand our proposed guidelines, it was apparent that familiarity is key to applying them in a given IoT application design in a short period of time. For our study, we printed the PbD guidelines on plain A4 sheets as a list. However, the experience of our study participants highlighted that this type of printed list is difficult to follow and can be more time consuming to use. We believe that approaches such as *Privacy Ideation Cards* [19] and KnowCards[6] would be more effective by allowing users to quickly familiarise themselves with the guidelines. In particular, using a colour coded, iconographic approach to represent the guidelines could help users remember them and thus leads to faster application of guidelines with less frustration.

An additional limitation of this work is that we did not consider the adaptive nature of privacy. While some decisions about implementing privacy preserving measures can be taken at design-time, IoT applications are by nature unpredictable. As a result, the ability to adapt is an important feature in IoT applications. Ideally, IoT applications should be able to compose built-in privacy preserving techniques into a run-time configuration, that maximises the privacy protection level while maintaining the overall utility of the application.

We would like to acknowledge that our design exercise is somewhat simplified compared to real-world industrial design. For example, most our participants omitted the latest distributed system design strategies such as Software Defined Networks (SDN) and Network Functions Virtualization (NFV). We would attribute this to a lack of specific knowledge on the part of our participants. However, we do not believe this issue impacts the results as our objective was to measure their PbD skills, not IoT infrastructure design skills.

7. Related Work

Our objective is to explore ways in which we can help software engineers to efficiently and effectively design privacy aware IoT applications. Towards this, in this paper, we proposed a Privacy-by-Design framework based on a set of guidelines and an associated method for applying them. There are a number of existing frameworks that have been proposed to help elicit privacy

[6] know-cards.myshopify.com

requirements and to design privacy capabilities into systems. Privacy principles, privacy strategies, privacy patterns have been developed to support software engineering processes. It is important to note that none of these approaches explicitly focus on the IoT domain or IoT application development processes.

Spiekermann [35] has identified a number of challenges in PbD approaches. Spiekermann identified PbD as *"an engineering and strategic management approach that commits to selectively and sustainably minimise information systems' privacy risks through technical and governance controls."*. Privacy is a vague concept without a rigid definition. Therefore, at times, it is difficult to measure the effectiveness or efficiency of privacy protection techniques. Further, distinguishing privacy from security is vital in order to develop methodologies to address privacy challenges. Spiekermann [35] also highlights the problem of not having widely agreed methodology for systematic engineering of privacy into systems. This justifies our attempt to develop a methodology to incorporate privacy protecting measures into IoT application designs.

Primarily, there are two approaches to incorporate privacy measures into a system design. The first approach is *threat-focussed*, explicitly examines a given system design to identify privacy threats and address them. LINDDUN [7], which we discuss later in this section, can be considered to be an example of a threat-focussed approach. Privacy Impact Assessment (PIA) [38] is also an example of this approach. The second approach is *threat-agnostic*, which proposes applying a series of privacy protecting measures to a given design without explicitly considering specific privacy threats. The expectation is to apply a set of blanket measures aiming to improve the overall privacy awareness of the design, not worrying about the threats involved. Our proposed methodology is an example of a threat-agnostic approach. Some other examples are privacy principles [5], and privacy strategies [13]. Both *'Threat-focussed'* and *'Threat-agnostic'* approaches have their own strengths and weaknesses. Due to the unique characteristics of each approach, a hybrid approach may create better system designs.

Threat-focussed This approach eliminates specific threats that a system might have. Therefore, it is a mission oriented approach where it forces system designers to think deeply about specific threats. On the down side, systems may struggle to handle threats that the designers haven't thought about at design time. Deep thinking processes would take more time and complexities could lead to poor threat analysis.

Threat-agnostic This approach is somewhat simpler and less error prone due to the absence of a threat analysis process. However, the same reason could lead to weak privacy design caused by not handling specific threats unique to a given system. On the other hand, this approach has more chance to handle unexpected privacy risks at run time due to lower dependence on threat identification processes. Therefore, highly dynamic systems may benefit from this approach.

Table 5: Summary of PbD Evaluation Methodologies

Area	Descriptions of Evaluation the approach
Garde-Perik [10]	This work explores the relative importance of complying with privacy related guidelines in the context of a Health Monitoring System. A total of 50 participants were given a text scenario describing a health care system. This system does not adhere to any of the OECD guidelines. Participants were then provided with potential fixes' to the system, each of which would make it comply with one specific OECD guideline. The guidelines were presented in pairs where participants needed to pick which guideline was most important.
Iachello et al. [15]	This work had developed a mobile application to conduct user studies in order to extract privacy guidelines. Those guidelines are then used to develop a second mobile application to evaluate and critique the proposed guidelines. Specific guidelines are presented later in this section.
Bellotti and Sellen [2]	This work has proposed a design framework for privacy in ubiquitous computing environments. They have proposed eleven criteria to evaluate a given design as presented later in this section. They take each criteria and evaluate it against their sample design.
LINDDUN [7]	LINDDUN is a threat modelling technique that supports the elicitation of privacy threats during the early stages of the software development life-cycle. Three groups have been involved in the evaluation process (total of 8 individuals) where they were asked to create a DFD diagram for a given high level scenario description (two groups focused on a e-health system and one group focused on a smart grid system) and use it to elicit the privacy threats using the LINDDUN framework. Group discussions were used to gather the participants' experience. They analysed both the results the participants documented in their reports (discovered threats), as well as the opinions of the participants with regard to their hands-on experience. • *Correctness:* On average, how many threats uncovered by the participants are correct (true positives vs false positives)? • *Completeness:* How many threats are undetected by the participants (false negatives)? • *Productivity:* How many valid threats are identified by the participants in a given time frame? • *Ease of use:* Did the participants perceive the methodology as easy to learn and apply? In order to explore any flaws in the LINDDUN method, the researchers asked a panel of three privacy experts to perform an independent threat analysis of a smart grid system using their own expertise. They have measured the reliability by comparing the expert designs with those of their study participants. • *Reliability:* Does LINDDUN miss any important threats?
Rubinstein and Good [32]	Based on a review of the technical literature, this work has derived a small number of relevant principles and illustrates them by reference to ten recent privacy incidents involving Google and Facebook.

Principles, Strategies, and Guidelines: The original PbD is a framework proposed by Ann Cavoukian [5], the former Information and Privacy Commissioner of Ontario, Canada. This framework identifies seven core principles by which privacy sensitive applications should be developed. These are: (1) proactive not reactive; preventative not remedial, (2) privacy as the default setting, (3) privacy embedded into design, (4) full functionality positive-sum, not zero-sum, (5) end-to-end security-full life-cycle protection, (6) visibility and transparency- keep it open, and (7) respect for user privacy, keep it user-centric. Cavoukian and Jonas [6] have extended these principles by proposing seven more specific guidelines to build PbD systems to manage big data, namely, (1) full attribution, (2) data tethering, (3) analytics on anonymized data, (4) tamper-resistant audit logs, (5) false negative favouring methods, (6) self-correcting false positives and (7) information transfer accounting. The ISO 29100 Privacy framework [16] has proposed eleven design principles, namely, (1) consent and choice, (2) purpose legitimacy and specification, (3) collection limitation, (4) data minimisation, (5) use, retention and disclosure limitation, (6) accuracy and quality, (7) openness, transparency and notice, (8) individual participation and access, (9) accountability, (10) information security, and (11) privacy compliance. Wright and Raab [40] has proposed to extend these ISO guidelines by adding 9 more guidelines, namely, (12) right to dignity, i.e., freedom from infringements upon the person or their reputation, (13) right to be let alone (privacy of the home, etc.), (14) right to anonymity, including the right to express one's views anonymously, (15) right to autonomy, to freedom of thought and action, without being surveilled, (16) right to individuality and uniqueness of identity, (17) right to assemble or associate with others without being surveilled, (18) right to confidentiality and secrecy of communications, (19) right to travel (in physical or cyber space) without being tracked, and (20) people should not have to pay in order to exercise their rights of privacy (subject to any justifiable exceptions), nor be denied goods or services or offered them on a less preferential basis.

The Fair Information Practice Principles (FIPPs) [4] proposed by the United States Federal Trade Commission is also formulated as set of guidelines, namely, (1) notice / awareness, (2) choice / consent, (3) access / participation, (4) integrity / security, and (5) enforcement / redress. Organisation for Economic Cooperation and Development (OECD) [39, 23] has also proposed similar privacy guidelines, namely, (1) notice, (2) purpose, (3) consent, (4) security, (5) disclosure, (6) access, and (7) accountability. Historically, OECD guidelines are considered as a successful milestone [39] where it laid the foundation for both subsequent Data Protection Directive (95/46/EC) and General Data Protection Regulation (GDPR) [8]. Rost and Bock [31] have identified six data protection goals, namely, (1) availability, (2) integrity, (3) confidentiality, (4) transparency, (5) unlinkability, and (6) ability to intervene. Fisk et al. [9] have proposed three privacy principles, namely, (1) least disclosure [internal disclosure, privacy balance, inquiry-specific release], (2) qualitative evaluation [legal constraints, technical limitations], and (3) forward progress.

Building on the ideas of engineering privacy by architecture vs. privacy-

by-policy presented by Spiekerman and Cranor [36], Hoepman [13] proposes an approach that identifies eight specific privacy design strategies: minimise, hide, separate, aggregate, inform, control, enforce, and demonstrate. This is in contrast to other approaches that we considered. In a similar vein, Singh et al. [34] have proposed 20 security considerations (somewhat similar to guidelines) for IoT systems, namely, (1) secure communications, (2) access controls for IoT-cloud, (3) identifying sensitive data, (4) cloud architectures: public, private, or hybrid?, (5) in-cloud data protection, (6) in-cloud data sharing, (7) encryption by *'things'*, (8) data combination, (9) identifying *'things'*, (10) identifying the provider, (11) increase in load, (12) logging at large scale, (13) malicious 'things'-protection of provider, (14) malicious 'things'-protection of others, (15) certification of cloud service providers, (16) trustworthiness of cloud services, (17) demonstrating compliance using audit, (18) responsibility for composite services, (19) compliance with data location regulations, and (20) impact of cloud decentralization on security.

Frameworks: LINDDUN [7] is a privacy threat analysis framework that uses data flow diagrams (DFD) to identify privacy threats. LINDDUN focuses on eliminating a set of pre-identified privacy threats using a systematic review of data flow diagrams. It consists of six specific methodological steps: (1) define the DFD, (2) map privacy threats to DFD elements, (3) identify threat scenarios, (4) prioritize threats, (5) elicit mitigation strategies, and (6) select corresponding privacy enhancing technologies. However, both LINDDUN and Hoepman's framework are not aimed at the IoT domain. Further, they not prescriptive enough in guiding software engineers. Bellotti and Sellen [2] have proposed a framework for design for privacy in ubiquitous computing environments. They argue that systems must be explicitly designed to provide feedback and control about (1) capture [when and what information collected], (2) construction [what happens to information], (3) accessibility [which people and what software have access to information], and (4) purposes [why data is being collected]. They also propose eleven criteria to evaluate a given design, namely, (1) trustworthiness, (2) appropriate timing, (3) perceptibility, (4) unobtrusiveness, (5) minimal intrusiveness, (6) fail-safety, (7) flexibility, (8) low effort, (9) meaningfulness, (10) learnability, (11) low cost. In contrast, the STRIDE [14] framework was developed to help software engineers consider security threats and is an example framework that has been successfully used to build secure software systems by industry. It suggests six different threat categories: (1) spoofing of user identity, (2) tampering, (3) repudiation, (4) information disclosure (privacy breach or data leak), (5) denial of service, and (6) elevation of privilege. However, its focus is mostly on security than privacy concerns.

8. Conclusions and Future Work

In this paper, we explored how a Privacy-by-Design (PbD) framework can help software engineers to design privacy-aware IoT applications. The PbD framework comprises a set of guidelines with a method for applying them and

we evaluated its effectiveness through an observational study where the participants were asked to design IoT applications to satisfy three given use cases. Our objective is to show how a set of guidelines can assist software engineers to design better privacy aware IoT applications. According to our findings, the proposed PbD framework significantly improved the privacy awareness of the IoT applications designed by both novice and expert software engineers. Further, our results show that software engineering expertise does not matter significantly when it comes to incorporating privacy protection features into IoT application designs. Finally, the qualitative data gathered during our studies highlighted a range of factors affecting privacy-aware IoT application design. These included different gaps in engineers' knowledge and understanding of privacy; and limitations in our approach that affected engineers' ability to apply the PbD guidelines effectively.

In the future, we will conduct research to develop a set of privacy tactics and patterns that are less abstract than guidelines. Such tactics and patterns will help software engineers to tackle specific privacy design challenges in IoT domain. At the moment, privacy guidelines are presented to the software engineers in plain text organised into a list. Though it is usable, in the future, we will explore how we can make these PbD guidelines more user friendly and accessible to software engineers. In particular, by using human-computer interaction techniques, we will help software engineers to efficiently and effectively browse and find relevant privacy guidelines, patterns and tactics in a given IoT application design context.

Acknowledgement

We acknowledge financial support for this work from the European Research Council (Advanced Grant 291652 - ASAP), EPSRC PETRAS 2 (EP/S035362/1) and EPSRC SAUSE (EP/R013144/1).

References

[1] Bass, L., Clements, P., Kazman, R., 2012. Software Architecture in Practice. 3 edition ed., Addison-Wesley Professional, Upper Saddle River, NJ.

[2] Bellotti, V., Sellen, A., 1993. Design for Privacy in Ubiquitous Computing Environments, in: Proceedings of the Third European Conference on Computer-Supported Cooperative Work 13-17 September 1993, Milan, Italy ECSCW '93, pp. 77–92.

[3] Carroll, J.M., Swatman, P.A., 2000. Structured-case: a methodological framework for building theory in information systems research. European Journal of Information Systems 9, 235–242.

[4] Cate, F.H., 2006. The Failure of Fair Information Practice Principles, in: Consumer Protection in the Age of the 'Information Economy', pp. 341–377.

[5] Cavoukian, A., 2009. Privacy by Design The 7 Foundational Principles Implementation and Mapping of Fair Information Practices. Technical Report. URL: https://www.iab.org/wp-content/IAB-uploads/2011/03/fred{_}carter.pdf.

[6] Cavoukian, A., Jonas, J., 2012. Privacy by Design in the Age of Big Data. Technical Report. Information and Privacy Commissioner, Ontario, Canada.

[7] Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W., 2011. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requirements Engineering 16, 3–32.

[8] European Commission, 2016. General Data Protection Regulation (GDPR). Official Journal of the European Union .

[9] Fisk, G., Ardi, C., Pickett, N., Heidemann, J., Fisk, M., Papadopoulos, C., 2015. Privacy principles for sharing cyber security data, in: Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015, pp. 193–197. doi:10.1109/SPW.2015.23.

[10] van de Garde-Perik, E., Markopoulos, P., de Ruyter, B., 2006. On the relative importance of privacy guidelines for ambient health care, in: Proceedings of the 4th Nordic conference on Human-computer interaction changing roles - NordiCHI '06, pp. 377–380. doi:10.1145/1182475.1182516.

[11] Gawande, A., 2011. The checklist manifesto : how to get things right.

[12] Haynes, A.B., Weiser, T.G., Berry, W.R., Lipsitz, S.R., Breizat, A.H.S., Dellinger, E.P., Herbosa, T., Joseph, S., Kibatala, P.L., Lapitan, M.C.M., Merry, A.F., Moorthy, K., Reznick, R.K., Taylor, B., Gawande, A.A., 2009. A Surgical Safety Checklist to Reduce Morbidity and Mortality in a Global Population. New England Journal of Medicine 360, 491–499. URL: http://www.nejm.org/doi/abs/10.1056/NEJMsa0810119, doi:10.1056/NEJMsa0810119.

[13] Hoepman, J.H., 2014. Privacy Design Strategies, in: Cuppens-Boulahia, N., Cuppens, F., Jajodia, S., Abou El Kalam, A., Sans, T. (Eds.), ICT Systems Security and Privacy Protection. Springer Berlin Heidelberg. volume 428 of *IFIP Advances in Information and Communication Technology*, pp. 446–459.

[14] Howard, M., Lipner, S., 2006. The security development lifecycle: SDL, a process for developing demonstrably more secure software. Microsoft Press.

[15] Iachello, G., Smith, I., Consolvo, S., Chen, M., Abowd, G.D., 2005. Developing privacy guidelines for social location disclosure applications and services, in: Proceedings of the 2005 symposium on Usable privacy and security - SOUPS '05, pp. 65–76. doi:10.1145/1073001.1073008.

[16] ISO/IEC 29100, 2011. Information technology Security techniques Privacy framework. Technical Report.

[17] Lewis-Beck, M., Bryman, A., Futing Liao, T., 2004. The SAGE Encyclopedia of Social Science Research Methods. Sage Publications, Inc., 2455 Teller Road, Thousand Oaks California 91320 United States of America.

[18] Lowrance, W., 2003. Learning from experience: privacy and the secondary use of data in health research. Journal of Health Services Research & Policy 8, 2–7.

[19] Luger, E., Urquhart, L., Rodden, T., Golembewski, M., 2015. Playing the Legal Card: Using Ideation Cards to Raise Data Protection Issues within the Design Process. Proceedings of the ACM CHI'15 Conference on Human Factors in Computing Systems 1, 457–466.

[20] Miles, M.B., Huberman, A.M., Saldaña, J., 2013. Qualitative data analysis : a methods sourcebook.

[21] Molich, R., Nielsen, J., 1990. Improving a human-computer dialogue. Communications of the ACM 33, 338–348. doi:10.1145/77481.77486.

[22] Nielsen, J., Molich, R., 1990. Heuristic evaluation of user interfaces, in: Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90, ACM Press, New York, New York, USA. pp. 249–256. doi:10.1145/97243.97281.

[23] Oleary, D.E., 1995. Some Privacy Issues in Knowledge Discovery: The OECD Personal Privacy Guidelines. IEEE Expert-Intelligent Systems and their Applications 10, 48–59. doi:10.1109/64.395352.

[24] Pearl, J., 1984. Heuristics : intelligent search strategies for computer problem solving. Addison-Wesley Pub. Co.

[25] Perera, C., Liu, C.H., Jayawardena, S., 2014a. A Survey on Internet of Things From Industrial Market Perspective. IEEE Access 2, 1660–1679.

[26] Perera, C., Liu, C.H., Jayawardena, S., 2015. The Emerging Internet of Things Marketplace from an Industrial Perspective: A Survey. IEEE Transactions on Emerging Topics in Computing 3, 585–598.

[27] Perera, C., McCormick, C., Bandara, A., Price, B., Nuseibeh, B., 2016. Privacy-by-design framework for assessing internet of things applications and platforms, in: ACM International Conference Proceeding Series. doi:10.1145/2991561.2991566.

[28] Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D., 2014b. Context Aware Computing for The Internet of Things: A Survey. Communications Surveys Tutorials, IEEE 16, 414 – 454.

[29] Richards, L., 2014. Handling qualitative data : a practical guide.

[30] Roman, R., Zhou, J., Lopez, J., 2013. On the features and challenges of security and privacy in distributed internet of things. Computer Networks 57, 2266–2279.

[31] Rost, M., Bock, K., 2011. Privacy by Design and the New Protection Goals. DuD, January , 1–9.

[32] Rubinstein, I.S., Good, N., 2013. Privacy by Design: A Counterfactual Analysis of Google and Facebook Privacy Incidents. Berkeley Technology Law Journal 28, 1333–1413. doi:10.2139/ssrn.2128146, arXiv:arXiv:1011.1669v3.

[33] Shi, E., Niu, Y., Jakobsson, M., Chow, R., 2011. Implicit Authentication through Learning User Behavior, in: Burmester, M., Tsudik, G., Magliveras, S., Ili, I. (Eds.), Information Security, Springer Berlin Heidelberg. pp. 99–113.

[34] Singh, J., Pasquier, T., Bacon, J., Ko, H., Eyers, D., 2016. Twenty Security Considerations for Cloud-Supported Internet of Things. IEEE Internet of Things Journal 3, 269–284. doi:10.1109/JIOT.2015.2460333, arXiv:1207.0203.

[35] Spiekermann, S., 2012. The challenges of privacy by design. Communications of the ACM 55, 38. doi:10.1145/2209249.2209263.

[36] Spiekermann, S., Cranor, L., 2009. Engineering Privacy. IEEE Transactions on Software Engineering 35, 67–82.

[37] Tashakkori, A., Teddlie, C., 2010. Sage handbook of mixed methods in social and behavioral research. SAGE Publications.

[38] Wright, D., De Hert, P., 2012. Privacy impact assessment.

[39] Wright, D., De Hert, P., Gutwirth, S., 2011. Are the OECD guidelines at 30 showing their age? Communications of the ACM 54, 119.

[40] Wright, D., Raab, C., 2014. Privacy principles, risks and harms. International Review of Law, Computers and Technology 28, 277–298. doi:10.1080/13600869.2014.913874.

[41] Xiong, W., Hu, H., Xiong, N., Yang, L.T., Peng, W.C., Wang, X., Qu, Y., 2014. Anomaly secure detection methods by analyzing dynamic characteristics of the network traffic in cloud communications. Inf. Sci. 258, 403–415. URL: https://doi.org/10.1016/j.ins.2013.04.009, doi:10.1016/j.ins.2013.04.009.

[42] Zhang, Q., Yang, L.T., Chen, Z., Li, P., Deen, M.J., 2018. Privacy-preserving double-projection deep computation model with crowdsourcing on cloud for big data feature learning. IEEE Internet of Things Journal 5, 2896–2903. doi:10.1109/JIOT.2017.2732735.